



TECHNICKÁ UNIVERZITA V LIBERCI  
Fakulta mechatroniky, informatiky  
a mezioborových studií ■

# DEMOÚLOHY PRŮMYSLOVÝCH ROBOTŮ ABB

## Bakalářská práce

*Studijní program:* B2612 – Elektrotechnika a informatika  
*Studijní obor:* 2612R011 – Elektronické informační a řídicí systémy  
*Autor práce:* **Jakub Hegner**  
*Vedoucí práce:* Ing. Josef Černohorský, Ph.D.





TECHNICAL UNIVERSITY OF LIBEREC  
Faculty of Mechatronics, Informatics  
and Interdisciplinary Studies ■

# DEMOTASKS FOR INDUSTRIAL ROBOTS ABB

## Bachelor thesis

*Study programme:* B2612 – Electrical Engineering and Informatics  
*Study branch:* 2612R011 – Electronic Information and Control Systems  
*Author:* **Jakub Hegner**  
*Supervisor:* Ing. Josef Černohorský, Ph.D.



## **ZADÁNÍ BAKALÁŘSKÉ PRÁCE**

**(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)**

**Jméno a příjmení:** Jakub Hegner  
**Osobní číslo:** M10000094  
**Studijní program:** B2612 Elektrotechnika a informatika  
**Studijní obor:** Elektronické informační a řídicí systémy  
**Název tématu:** Demoúlohy průmyslových robotů ABB  
**Zadávací katedra:** Ústav mechatroniky a technické informatiky

### **Z á s a d y   p r o   v y p r a c o v á n í :**

1. Navrhněte alespoň dvě vhodné úlohy pro potřeby prezentace možností robotů ABB IRB 120 a IRB 140.
2. Realizujte model jedné z úloh v prostředí RobotStudio.
3. Vyřešte rychlou přípravu úlohy, aretačním systémem či softwarovou korekcí.
4. Ověřte vámi navržené řešení.

Rozsah grafických prací: dle potřeby dokumentace

Rozsah pracovní zprávy: 30–40 stran

Forma zpracování bakalářské práce: tištěná/elektronická

Seznam odborné literatury:

- [1] Technická dokumentace ABB RapidLanguage, ABB RobotStudio
- [2] Skařupa J.: Průmyslové roboty a manipulátory, VŠB-TUO Ostrava 2007, ISBN 978-80-248-1522-0

Vedoucí bakalářské práce:

**Ing. Josef Černohorský, Ph.D.**

Ústav mechatroniky a technické informatiky

Konzultant bakalářské práce:

**Ing. David Lindr, Ph.D.**

Ústav mechatroniky a technické informatiky

Datum zadání bakalářské práce: 10. října 2013

Termín odevzdání bakalářské práce: 16. května 2014

  
prof. Ing. Václav Kopecký, CSc.  
děkan



  
doc. Ing. Milan Kolář, CSc.  
vedoucí ústavu

V Liberci dne 10. října 2013

## Prohlášení

Byl jsem seznámen s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu TUL.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Bakalářskou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím mé bakalářské práce a konzultantem.

Současně čestně prohlašuji, že tištěná verze práce se shoduje s elektronickou verzí, vloženou do IS STAG.

Datum: 16.05.2014

Podpis: 

## **Poděkování**

Rád bych poděkoval vedoucímu mé Bakalářské práce Ing. Josefu Černohorskému, Ph.D. za rady, věcné připomínky, odborný dohled a především trpělivost. Za totéž bych chtěl poděkovat i Ing. Davidu Lindrovi, Ph.D. Dále bych rád poděkoval Jaroslavu Habadovi za pomoc a při výrobě kovové trasy a nástroje.

## Abstrakt

Tato Bakalářská práce se zabývá především výrobou a návrhem úloh vhodných pro prezentaci průmyslových robotů firmy ABB, které by mohla využít Technická universita v Liberci – konkrétně se týká robotů ABB IRB 120, ABB IRB 140 a ABB IRB 1400 a jejich řídicích systémů S4C plus a IRC5. V úvodu práce v kapitole 2 se věnuji vybavení a technických prostředcích dostupných v laboratoři robotiky Technické univerzity v Liberci umožňujících vypracování oněch úloh. Dále, v kapitole 3 budete informováni o návrzích úloh „Horký drát“ a „Předání míčku“, kde se mimo jiné dozvíte, jakou podobu tyto úlohy mají a z jakých důvodů byly zvoleny právě tyto. Taktéž se v této části zabývám návrhem a výrobou prostředků a nástrojů, potřebným k jejich vypracování. V neposlední řadě (kapitola 4) se dozvíte o vytváření zmíněných úloh v programovém prostředí RobotStudia včetně prostředků, které program RobotStudio poskytuje. Je zde též zhodnocena simulace a zdůvodněn postup tvorby úloh. Závěrem je nastíněna aretace a problémy, jež se během procesu vytváření inkriminovaných úloh vyskytly.

**Klíčová slova:** návrh, ABB, roboty, RobotStudio, simulace.

## **Abstract**

This Bachelor thesis is focused mainly on designing and developing tasks suitable for presentation of industrial robots from ABB company which can use Technical University of Liberec – especially robots ABB IRB 120, ABB IRB 140 a ABB IRB 1400 and their control systems S4C plus and IRC5. In the first paragraphs you can find something about equipment and technical means available in laboratory of robotics in Technical University of Liberec which enable developing of these tasks. Further in chapter 3 on you will be informed about design of tasks "Hot wire" and "Passing of ball" where you will get to know, besides other things, how these tasks look like and why were these tasks chosen. In this part, you can also find something about designing and developing means and tools which are needed for creating these tasks. Last but not least (chapter 4) you will get to know about developing these tasks in programmable environment RobotStudio including possibilities which program RobotStudio provides. In the thesis there is also an evaluated simulation and reasons of chosen procedures are given. In conclusion there also are shown detent and issues which appeared during the process of developing the tasks.

**Keywords:** design, ABB, robots, RobotStudio, simulation.



# Obsah

<b>Poděkování.....</b>	<b>6</b>
<b>Abstrakt.....</b>	<b>7</b>
<b>Abstract.....</b>	<b>8</b>
<b>Obsah .....</b>	<b>9</b>
<b>Seznam obrázků.....</b>	<b>11</b>
<b>1 Úvod .....</b>	<b>12</b>
<b>2 Technické prostředky pro realizaci úloh .....</b>	<b>13</b>
2.1 Laboratoř.....	13
2.2 Roboty .....	13
2.3 Řídící systém (Robot Controller) .....	15
2.4 Efektory .....	15
2.5 Odkládání nástrojů .....	17
2.6 Programové prostředí – RobotStudio .....	18
<b>3 Návrh úloh .....</b>	<b>19</b>
3.1 Úloha 1: Horký drát .....	19
3.1.1 Nástroj.....	20
3.1.2 Trasa.....	22
3.1.3 Obvod pro detekci dotyku.....	23
3.2 Úloha 2: předávání míčku .....	25
<b>4. RobotStudio .....</b>	<b>27</b>
4.1 Úloha 1: Horký drát .....	27
4.1.1 Efektor.....	27
4.1.2 Nástroj.....	28
4.1.3 Tvorba workobjectů .....	29
4.1.4 Tvorba trasy .....	30
4.1.5 Uchopení a odložení nástroje .....	32
4.1.6 Události .....	33
4.1.7 Dokončení programu .....	34
4.1.8 Zkouška simulace .....	34
4.2 Úloha 2: Předání míčku .....	34

4.2.1 Efektory .....	34
4.2.2 Workobjecty .....	35
4.2.3 Tvorba cesty .....	35
4.2.4 Uchopení a odložení míčku .....	36
4.2.5 Předání míčku .....	36
4.2.6 Události .....	37
4.2.7 Dokončení programu .....	37
4.2.8 Zkouška simulace .....	37
<b>5 Konečné doladění .....</b>	<b>38</b>
5.1 Aretační systém a ukotvení .....	38
5.2 Problémy .....	38
<b>Závěr .....</b>	<b>40</b>
<b>Literatura .....</b>	<b>42</b>
<b>A Obsah CD .....</b>	<b>44</b>

## Seznam obrázků

1.	ABB IRB 140 .....	13
2.	ABB IRB 1400 .....	14
3.	ABB IRB 120 .....	14
4.	Adaptivní chapadlo FESTO .....	16
5.	Graf závislosti tuhosti pružiny na změně její délky .....	18
6.	Chapadlo od firmy Festo.....	20
7.	Vytvořený nástroj .....	20
8.	Efektor držící elektrodu .....	21
9.	Trasa <i>Horkého drátu</i> .....	22
10.	Navařená vrchní podložka .....	23
11.	Přidělaný konce trasy za pomoci závitů a matky .....	23
12.	Patice a relé Finder .....	23
13.	Zapojení obvodu .....	24
14.	Krabice s umístěnými součástkami .....	25
15.	Vytvoření modelu nástroje .....	27
16.	Poslední krok vytváření efektoru .....	28
17.	Rozdíl mezi modelem a skutečnou podobou nástroje .....	29
18.	Namodelovaná trasa .....	31
19.	Nejzajímavější část trasy .....	32
20.	Vytvořená virtuální jednotka .....	33
21.	Event Manager k úloze 1 .....	33
22.	Event Manager k úloze 2 .....	37

# 1 Úvod

V dnešní době je robotika oborem, jež má stále velký prostor pro vývoj a modernizaci, ať už se jedná o její využití v běžné domácnosti (kuchyňské roboty, robotické vysavače, ...), či nespočet možností využití v průmyslu. V nejvyšší míře jsou roboty využívány ve výrobě s velkou sériovostí, kde zásadně urychlují a zpřesňují celý výrobní proces. Jedná se tedy o pokrokové, perspektivní odvětví s velkou možností uplatnění a mělo by tedy být dostatečnou motivací pro uchazeče o studium na Technické univerzitě v Liberci na Fakultě mechatroniky, informatiky a mezioborových studií stejně tak, jako bylo vyhotovení této práce motivací pro mě.

Konkrétní práce má za účel demonstrovat možnosti průmyslových robotů ABB dostupných v laboratoři Robotiky na TU v Liberci. Jednak by mělo být možné jednotlivé úlohy jednoduše a bez náročné přípravy použít během dnů otevřených dveří jakožto část studia, se kterou se uchazeč v budoucí výuce setká – jednak při samotné výuce pro prezentaci dosahu a rozsahu jednotlivých os robotu, případně i vzájemné propojení s programem RobotStudio.

## 2 Technické prostředky pro realizaci úloh

### 2.1 Laboratoř

Na základě projektu FRVŠ byly v roce 2004 zakoupeny tři průmyslové roboty od firmy ABB včetně řídicích systémů (roboty IRB 140 a IRB 1400 s řídicími systémy S4C+ a 1ks repasovaného robota IRB1400 M97A s řídicím systémem S4C). Součástí dodávky bylo též 12 ks vizualizace Robot studio pro výuku, umožňujících vytvářet programy odděleně od robotů s možností přímé verifikace simulací na PC. V roce 2010 bylo zakoupeno čtvrté robotické rameno od firmy ABB (IRB 120). [6]

### 2.2 Roboty

Při této práci byly využity tři ze čtyř dostupných robotů ABB v laboratoři Robotiky na Technické univerzitě v Liberci. Mimo zadaných robotů IRB 120 a IRB 140 byl použit jeden robot IRB 1400 pro potřeby druhé úlohy, jelikož roboty IRB 120 a IRB 140 nejsou ve vzájemném dosahu.

- IRB 140 – Tento průmyslový robot je kompaktním a výkonným šestiosým strojem s unikátní kombinací velkého zrychlení, odolnosti, pracovního dosahu a užitečného zatížení. Vyznačuje se též velkou střední dobou bezporuchového provozu a velmi vysokou opakovatelností. Díky užitečnému zatížení do 6kg a dosahu 810mm je tento průmyslový robot nejlepší ve své třídě. [8]



1. ABB IRB 140

- IRB 1400 – Je známý pro svou tuhost, robustnost, velký pracovní prostor a dlouhý dosah. Je dostatečně rychlý a spolehlivý. Osvědčil se především v obloukovém svařování, manipulacemi s materiálem či procesními aplikacemi. Dokáže manipulovat s předměty vážícími až 5kg, přičemž je schopen na horní části paži nést doplňkovou zátěž o hmotnosti až 18kg. [9]



## 2. ABB IRB 1400

- IRB 120 – Nejmenší víceúčelový průmyslový robot společnosti ABB váží pouhých 25kg a dokáže manipulovat s břemenem o hmotnosti až 3kg (4kg ve svislé ose) s dosahem ramene 580 mm. Jedná se o úspornou a spolehlivou volbu s vysokou výrobní kapacitou a nízkými pořizovacími náklady. [7]



## 3. ABB IRB 120

## 2.3 Řídící systém (Robot Controller)

V laboratoři robotiky se vyskytují roboty s následujícími systémy:

- S4C – Jedná se o stále používaný funkční systém, který je dostatečně spolehlivý a je možno jej používat mnoha prostředích. Ve školní laboratoři ho používá repasovaný robot IRB1400 M97A. Velkou nevýhodou tohoto systému je disketová jednotka, která je jedinou možností pro export/import dat z/do systému. [10]
- S4C plus – Inovovaná verze systému S4C, která umožňuje komunikaci přes ethernet, případně flash disk, či sériovou linku. Stejně tak jako s prací se systémem S4C je dobré dát si pozor na dlouhé názvy targetů či workobjectů. V případě, že je program vytvořen v současných verzích programu RobotStudio (5.14, 5.15, 5.60, 5.61) je třeba jej pro správné spuštění ručně upravit. FlexPendant se ovládá pouze pomocí tlačítek a není barevný, tudíž přehlednost, orientace a celková práce s ním nedosahuje tak vysokého standartu jako v případě IRC5. [11]
- IRC5 – Tento systém v první řadě umožňuje interaktivní propojení řídicího systému s programem. FlexPendant je dotykový, což velice zjednodušuje a urychluje průběh celého programování, popřípadě úprav programu. Barevné prostředí též napomáhá čitelnosti, přehlednosti a celkové orientaci v programu. [12]

Z hlediska urychlení programování, přehlednosti a nespočtu dalších výhod, které poskytuje řídicí systém IRC5 by bylo vhodné mít k dispozici tento systém pro všechny roboty ABB dostupné v laboratoři Robotiky, nicméně i použití starších má své výhody. Ne všechny firmy jsou vybaveny nejmodernější technikou a tak je třeba, aby byli studenti připraveni i na variantu, že se mohou v praxi setkat se starší technikou.

## 2.4 Efektory

V laboratoři dostupné přísavky se skvěle hodí pro realizaci úlohy „Předání míčku“. Pro realizaci druhé jsou k dispozici chapadla s dvěma, respektive třemi uchopovacími body. Vzhledem k tomu, že byla potřeba co nejpřesnější a nejstálější

pozice uchopeného předmětu (nástroje), bylo využito chapadlo s třemi uchopovacími body, které se opírá o tvarový i silový stisk. Nevýhodou tohoto chapadla je především maximální sevření jeho čelistí. V případě použití šestihranu je potřeba aby vzdálenost jeho protějších stěn byla alespoň 22mm, v opačném případě čelisti šestihran nesevrou. Vhodným řešením by bylo například vyhotovení jiných čelistí, aby mohli být tímto efektozem uchopovány i předměty o menších rozměrech.

Pokud by se rozšířila škála dostupných efektozů v laboratoři, bylo by možné vytvářet mnohem zajímavější úlohy. Zároveň by mohli být roboty prospěšné i pro práce na jiných fakultách Technické university v Liberci. Nebylo by špatné mít k dispozici například magnetický úchopový prvek s elektromagnetem, zajímavé by bylo především jeho porovnání s přísavkou, ale i s nyní dostupným chapadlem a to nejen v možnosti reálného použití, ale i v náročnosti na vytvoření programu pro realizaci onoho užití.

Dalším užitečným a zajímavým prvkem, který by přispěl k inovaci a urychlení práce laboratoře robotiky, by zajisté byl systém pro rychlou výměnu efektozů. Dostatečná by byla redukční část pro rychlou výměnu. Nicméně, prostředky dostupné v laboratoři jsou pro tuto práci a pro první seznámení s roboty ve výuce dostačující.

V dnešní době se rozšiřuje používání adaptivních chapadel (viz obrázek 4), díky nimž odpadá potřeba efektozy měnit, avšak tato varianta je stále velice finančně náročná. Finanční náročnost roste lineárně s adaptibilitou daného chapadla, jakožto rostou i náklady na výrobu chapadla.



4. Adaptivní chapadlo FESTO [13]



## 2.5 Odkládání nástrojů

V laboratoři není k dispozici žádný podstavec na odkládání nástrojů, a tudíž bylo třeba jej vyrobit. V ideálním případě by byl odpružen, aby nedocházelo k deformaci nástroje, podstavce, či pracovní desky, na které je podstavec umístěn. To se může projevit při odkládání nástroje na podstavec.

Při hmotnosti nástroje o maximální hmotnosti 15kg a maximální požadované změny délky pružiny o 1mm by koeficient tuhosti dané pružiny odpovídal přibližně hodnotě  $k_c \approx 147,1 \text{ kN} \cdot \text{m}^{-1}$ , což vyplývá z následujícího vyjádření (1).

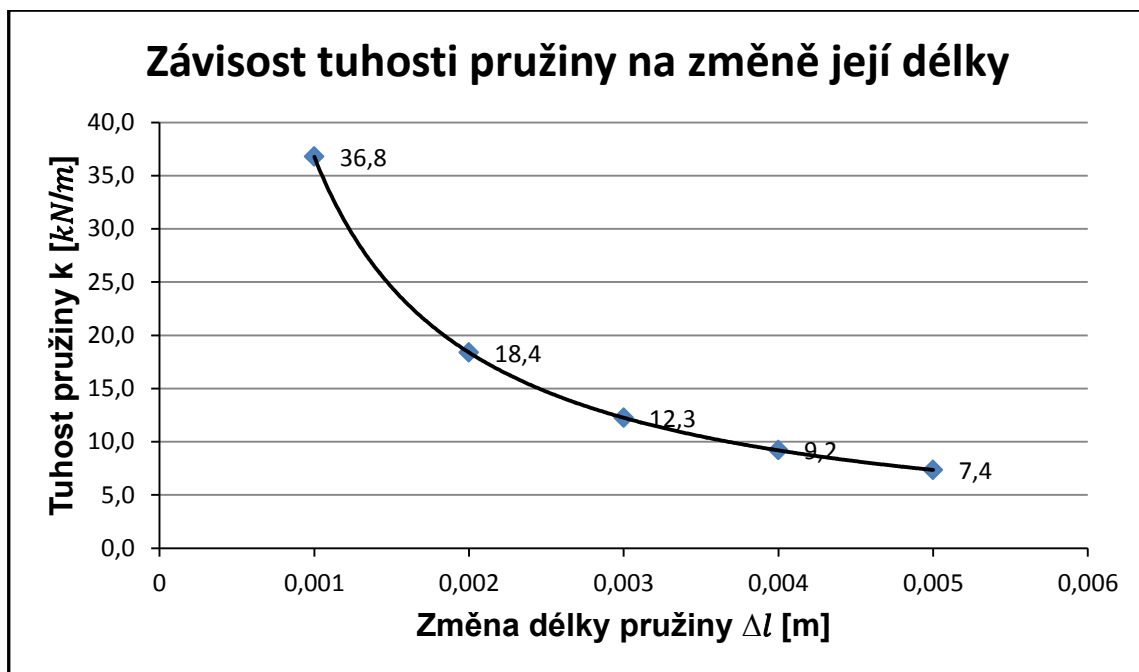
$$k_c = \frac{F}{\Delta l} = \frac{m \cdot a}{\Delta l} = \frac{m \cdot g}{\Delta l} = \frac{15 \cdot 9,80665}{0,001} = 147099,75 \text{ N} \cdot \text{m}^{-1} \approx 147,1 \text{ kN} \cdot \text{m}^{-1} \quad (1)$$

Uvážíme-li, že podstavec má čtyři nohy a pružina bude tedy na každé z těchto noh, vydělíme výsledek čtyřmi a získáme tak tuhost, kterou potřebujeme u každé této jednotlivé pružiny.  $k_j \approx 36,8 \text{ kN} \cdot \text{m}^{-1}$  (2).

$$k_j = \frac{k_c}{4} = \frac{147099,75}{4} = 36774,9375 \text{ N} \cdot \text{m}^{-1} \approx 36,8 \text{ kN} \cdot \text{m}^{-1} \quad (2)$$

Případ, kdy bychom chtěli jinou změnu délky pružiny  $\Delta l$ , je uveden v následujícím grafu (5).

V grafu lze vidět, že závislost mezi změnou délky pružiny a tuhosti pružiny není lineární. Tuhost pružiny je zde udávána v jednotkách kN/m změna délky pružiny pak v m. Hodnoty jsou počítány pro jednu pružinu z celkových čtyř – pokud tedy uvažujeme, že podstavec pro odkládání nástrojů bude mít čtyři nohy.



5. Graf závislosti tuhosti pružiny na změně její délky

## 2.6 Programové prostředí – RobotStudio

RobotStudio, je program, který umožňuje offline programování, na které je v průmyslu kladem velký důraz např. proto, že při vytváření nebo úpravě programů není třeba zastavovat provoz reálného robotu.

RobotStudio běží paralelně s virtuálním kontrolerem. Tyto programy spolu sice spolupracují, ale fungují nezávisle na sobě. Programy, jež v nich vytváříme, můžeme vytvářet v jednom či v druhém programu. Též je možné část vytvářet v RobotStudiu a část ve virtuálním kontroleru, ovšem je třeba obezřetnosti, aby nedošlo k přepsání jedné či druhé části.

Ke vzájemnému provázání mezi oběma programy nedochází automaticky. V případě, že chceme program vytvořený v RobotStudiu nahrát do virtuálního kontroleru, používá se funkce *Synchronize to VC*. V případě, že chceme opak, použijeme funkci *Synchronize to Station*.

RobotStudio dále umožňuje modelovat jednoduchou geometrii, ale opravdu jen jednoduchou. V případě složitějších modelů je mnohem lepší vytvořit modely například v programu AutoCad a následně je do RobotStudia importovat. Dobře využitelnou funkcí RobotStudia je i tvorba nástroje.

## 3 Návrh úloh

Obě úlohy byly vybrány z více návrhů po konzultaci s vedoucím a konzultantem práce. V jejich návrhu byla zohledněna mimo jiné potřeba Technické univerzity, aby co nejvíce vyhověly technickému zadání a splňovaly potřeby laboratoře.

### 3.1 Úloha 1: Horký drát

Úloha, jak již název napovídá, vychází ze hry nazývané Horký drát. Cílem této hry je projet kovovou trasu za pomoci kovového očka aniž by došlo k jejich vzájemnému dotyku. Úloha založená na stejném principu je sice již pro potřeby prezentace na dny otevřených dveří vytvořena, nicméně není dostatečně dlouhá, odolná a vzhledem k tomu, že je vyrobena z velice měkkého materiálu, je nestálá, proto byla po dohodě s vedoucím práce zvolena varianta na přepracování této úlohy do robustnější podoby. Na rozdíl od původní, je tato úloha vytvořena ve 3D a je rozměrově mnohem větší. Velikost v tomto případě hraje důležitou roli pro případného diváka, jelikož mnohem lépe uvidí celé provedení, ale zároveň se mnohem lépe zdůrazní rozsahové možnosti robotu. Trojrozměrné provedení trasy zase klade důraz na rozsahové možnosti os robotu.

V tomto případě je kovová elektroda ve tvaru „C“ držena robotem ABB IRB 120 za pomoci tříbodového chapadla, které je dostupné v laboratoři robotiky na Technické univerzitě v Liberci a můžete jej vidět na obrázku 6. Samostatný výrobek firmy Festo však nestačí, jsou na něm přimontovány jednotlivé prsty ve tvaru písmene „L“, které po stlačení drží nástroj. Tyto pacičky byly dodatečně vyrobeny na TU v Liberci. Sevření čelistí chapadla má pak na starosti stlačený vzduch.

Případný vzájemný dotyk je detekován vytvořeným jednoduchým elektrickým obvodem, jehož vývody jsou přivedeny na elektrodu a kovovou trasu „Horkého drátu“.



6. Chapadlo od firmy Festo [16]

### 3.1.1 Nástroj

Samotný nástroj (viz obrázek 7) je vytvořen ze dvou kovových částí, rukojeti a oka. Je u něj kladen důraz především na pevnost, stálost a v neposlední řadě by měl umožňovat své držení v neměnné pozici.



7. Vytvořený nástroj

První část – rukojeť – je ocelová tyč s průřezem pravidelného šestibokého hranolu o výšce 120mm. V tomto případě se ideálně hodí, jelikož sevřením chapadel se dokonale zajistí přesně vystředěná pozice nástroje. Zároveň není možné, aby byl nástroj

nijak poškozen velkým tlakem vzduchu, který zajišťuje jeho držení čelistmi. Jeho přesné uchopení je vidět na obrázku 8.



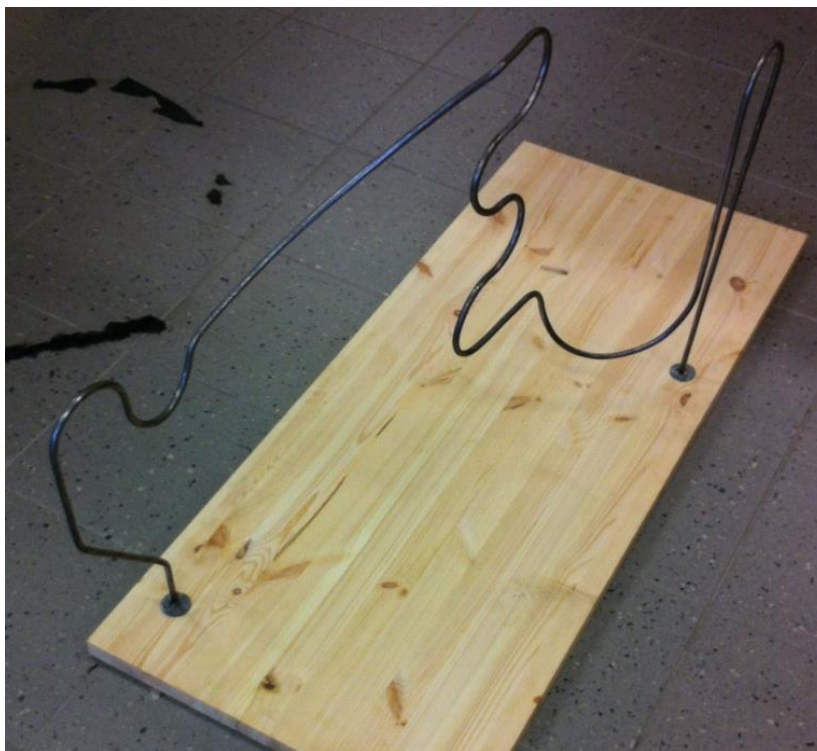
#### 8. Efektor držící elektrodu

Druhá část – oko – je vytvořeno z běžně dostupné ocelové kulatiny o průřezu 8mm. Kulatina je v nejvzdálenějším bodě přerušena. Důvodem pro toto přerušení je skutečnost, že v opačném případě by byla nutná manipulace obsluhy k nasazení nástroje do pozice vhodné k projetí labyrintu. V tomto případě je robot schopen po uchopení nástroje sám najet na start trasy a na konci trasy opustit a vrátit nástroj zpět na odkládací podstavec. Tímto krokem se tedy urychlí celý proces instalace úlohy. Kulatina byla ohnuta do výsledné pozice v domácí dílně okolo víčka vhodného rozměru, které mělo kruhový půdorys. Posléze byl její tvar doladěn ve svěráku, jelikož měl materiál tendenci se vracet do výchozí pozice a tedy se rozvírat.

Obě výše zmíněné části byly následně svařeny za pomoci CO<sub>2</sub>, což je relativně dostupná, a rychlá metoda sváření, u níž nevzniká žádná struska. Navíc jsou sváry provedené touto metodou relativně hezké na pohled. Navíc, k ještě lepší vizuální podobě byl svár lehce opracován úhlovou bruskou – *flexou*, aby byl následně jemně přebroušen pilníkem na kov. Toto přebroušení nevylepšilo jen vizuální stránku, ale především bezpečnost, jelikož došlo ke stržení a také zaoblení ostrých hran, které by mohli způsobit nechtěná poranění. Hrany byly zaoblény též v místě, kde je přerušeno oko.

### 3.1.2 Trasa

Celá trasa (viz obrázek 9) je vytvořena z totožného materiálu jako oko nástroje, tedy kulatiny o průřezu 8mm. Její ohýbání a celkové dotvarování bylo opět vytvářeno v domácí dílně za pomoci předmětů vhodných pro realizaci ohybů, tentokrát především kruhového půdorysu o průměru 100 respektive 200mm. Konečné dotvarování se realizovalo za pomoci svěráku.

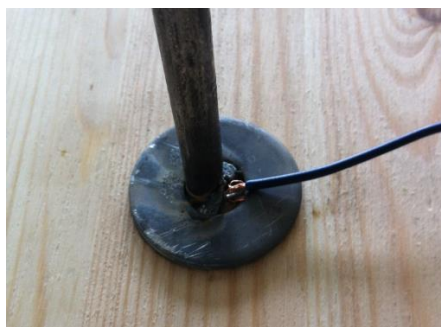


9. Trasa *Horkého drátu*

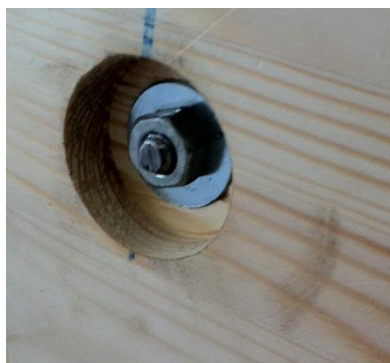
Trasa je vytvořena ze dvou kusů této ocelové kulatiny, jelikož je nejdostupnější k sehnání v délce 2m, ovšem délka trasy je přibližně 3,5m. Byly tedy vytvořeny dvě části, které se následně svařily, a opracovali k podobě, kdy je svár naprosto neznatelný. Trasa byla vyhotovena dle předlohy, vymodelované v programu RobotStudio od společnosti ABB (více v kapitole 4).

Na oba konce byly též navařeny podložky, aby mohla celá konstrukce stabilně držet ve svém podkladu, v tomto případě 28mm široké desce z masivního dřeva o rozměrech 50x120mm. Do desky byly následně vyvrtány dva otvory. Dále byly oba konce kulatiny opatřeny metrickým závitem, aby bylo možné je ze spodní strany desky přitáhnout matkami k již navařeným podložkám a celá konstrukce tak pevně držela na desce, což lze vidět na obrázcích 10 a 11.





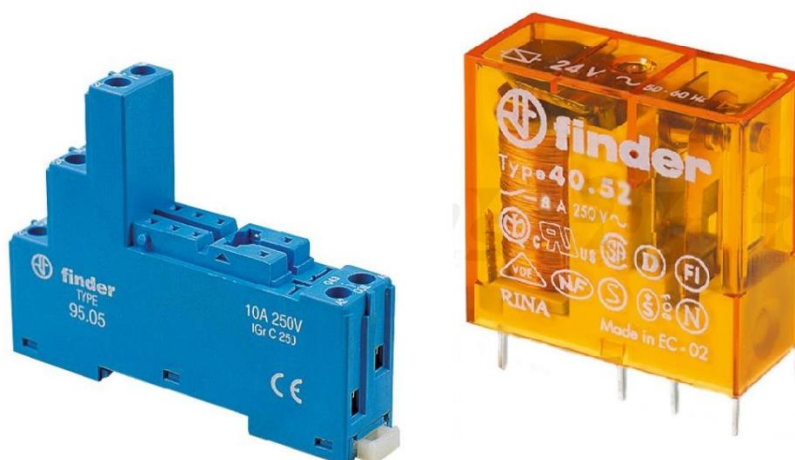
10. Navařená vrchní podložka



11. Přidělaný konce trasy za pomoci závitu a matky

### 3.1.3 Obvod pro detekci dotyku

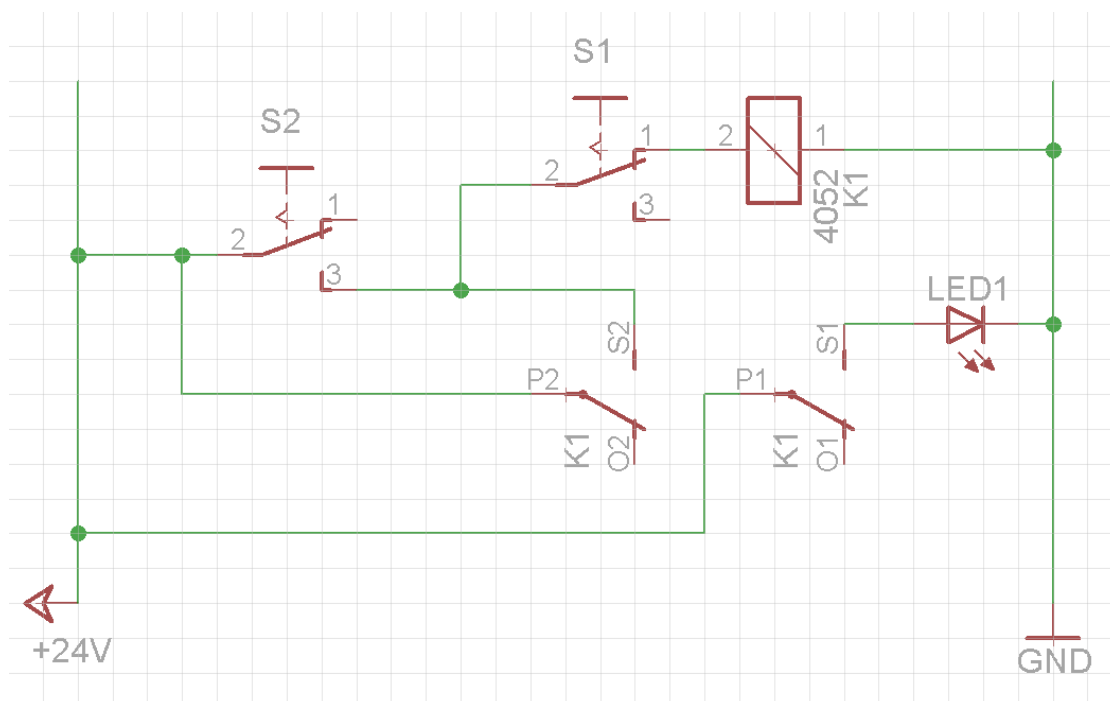
Poslední, jež zbývalo k fyzické kompletaci této úlohy, byl elektrický obvod, který má za úkol signalizovat případný dotyk mezi elektrodou a trasou. Hlavní tíhu celého obvodu nese přepínací relé Finder 40.52 s paticí Finder 95.05 - viz obrázek 12, pro jehož přepnutí je potřeba napětí 24V.



12. Patice a relé Finder [14][15]

Napětí zdroje, který je použit v daném obvodu je možno jednoduše regulovat, ale kvůli výše zmíněnému přepínacímu napětí relé je zvoleno jeho maximální napětí – tedy 24V.

Dále je použita dioda, která signalizuje dotyk, tlačítko pro rozpojení obvodu a několik vodičů. Zapojení obvodu je vidět na obrázku č. 13, který je vytvořen v programu Eagle.

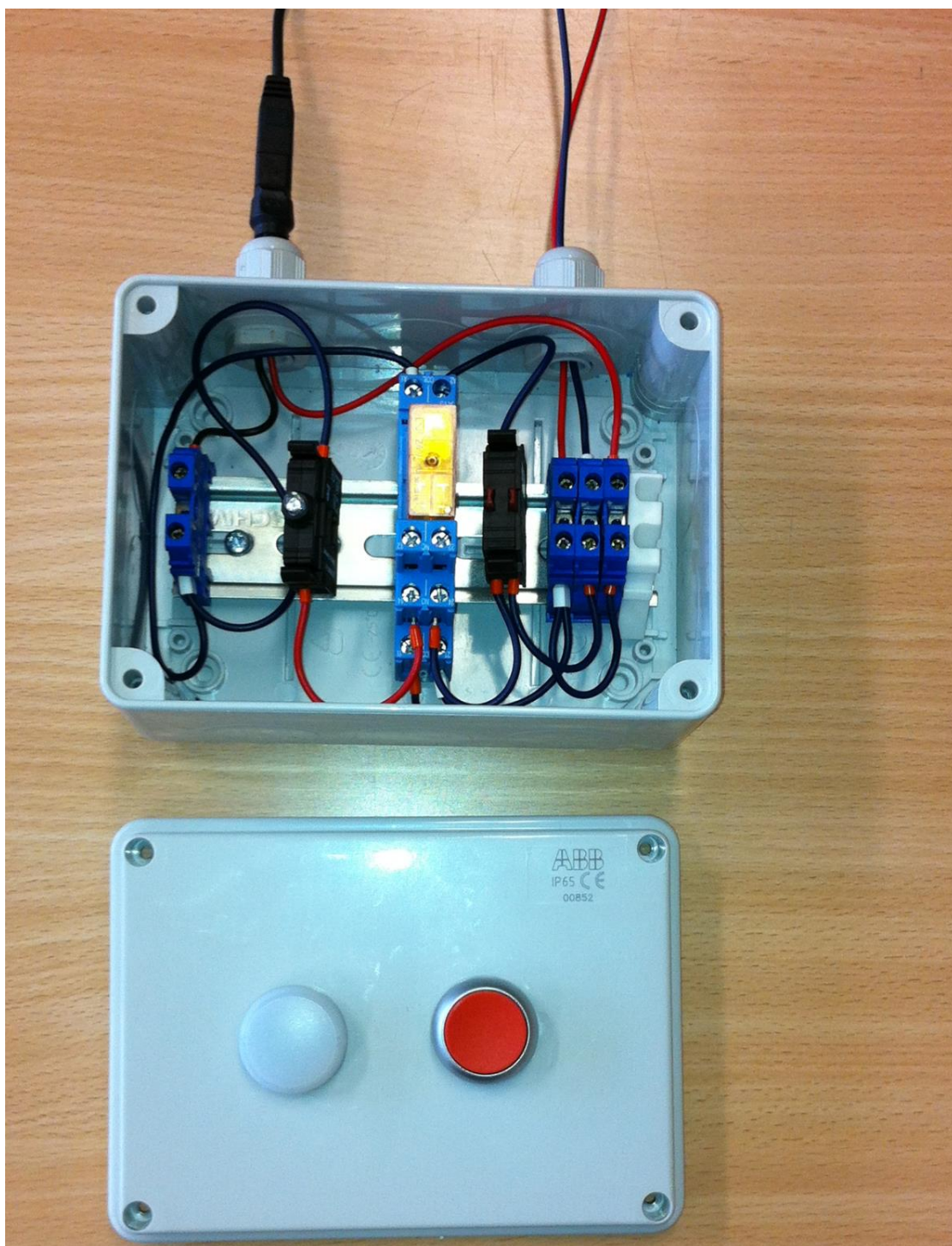


13. Zapojení obvodu

Namísto elektrody a trasy, ke kterým je obvod připojen, je v zapojení použito tlačítko S2, jenž simuluje jejich vzájemný kontakt. Tlačítko S1 je reálné tlačítko sloužící pro rozpojení obvodu.

Celý obvod je umístěn do elektroinstalační plastové krabice (obrázek 14), aby vypadal úhledně. Signálka i tlačítko jsou vyvedeny na vrchní krytku pro snadné zacházení. Konektor pro připojení napájení a vedení pro připojení na elektrodu a trasu jsou vyvedeny pomocí průchodek na zadní stranu krabice. Drátky jsou pak jednoduše připájeny – jeden k elektrodě, druhý k trase. Svorky a patice relé jsou připojeny na din lištu, která je přišroubovaná ke dnu krabice. Celá krabice je umístěna a přišroubována na okraj podkladové desky.





14. Krabice s umístěnými součástkami

### 3.2 Úloha 2: předávání míčku

V této úloze si budou dva roboty mezi sebou vyměňovat míček. Oproti původnímu předpokladu, kdy měli spolupráci vykonávat roboty ABB IRB 120 a ABB IRB 140, bylo nutné nastolit změnu, jelikož zmíněné roboty nebyly ve vzájemném

dosahu a bez jejich nutné manipulace by úloha nebyla proveditelná. Robot ABB IRB 140 byl v úloze zachován a jako druhý se použil ABB IRB 1400. Oba tyto roboty využívají řídicí systém S4C plus.

Úloha je zaměřena především na spolupráci dvou robotů. Jako efektory jsou použity dvě přísavky.

Jeden z robotů nejprve najede do pozice pro převzetí míčku, posléze míček „uchopí“ přísavkou a najede do pozice pro předání. Současně s tímto krokem najede do pozice též druhý robot. Dojde k výměně míčku, kterou mají na starosti přísavky a oba roboty se vrátí do výchozí pozice, přičemž druhý robot míček odloží na určeném místě. Tento cyklus se posléze opakuje v opačném gardu.

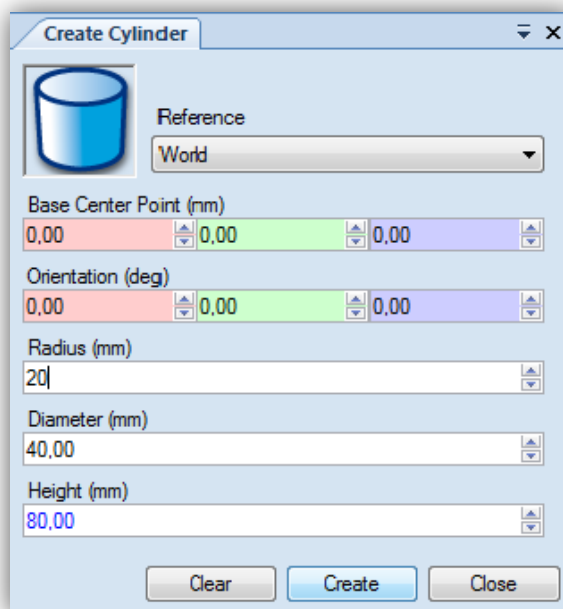
## 4. RobotStudio

### 4.1 Úloha 1: Horký drát

Dle zadání měla být jedna z úloh vytvořena zároveň v prostředí RobotStudio. Zvolil jsem variantu, že touto úlohou bude Horký drát. Pro uživatele jednoduchá komunikace RobotStudia a řídicího systému IRC5 k tomu ostatně vybízí a vzhledem k tomu, že systém IRC5 je používán k řízení robotu ABB IRB 120.

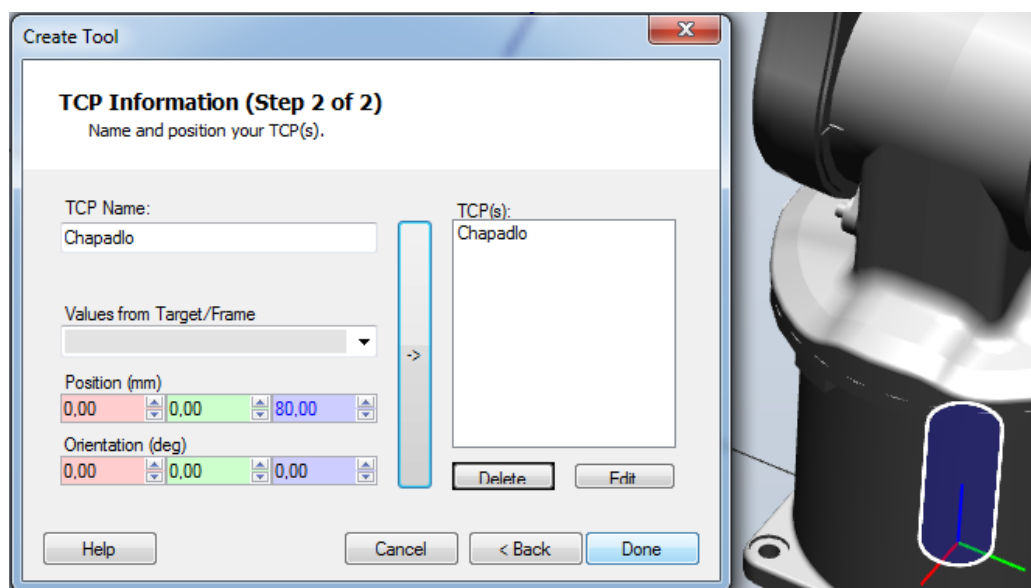
#### 4.1.1 Efektor

Po založení projektu s odpovídajícím robotem bylo potřeba v první řadě vytvořit objekt, který by simuloval reálný efektor. V našem případě je naprosto dostačující namodelovat válec o shodné výšce, jako je výška reálného efektoru – tří bodového chapadla (viz obrázek 15). Změna orientace ani pozice není nutná, naopak je pro vytváření nástroje lepší zachovat obě beze změny.



15. Vytvoření modelu nástroje

Z hotového modelu je již možné vytvořit samotný efektor, určí se pouze název, tělo, ze kterého se má efektor vytvořit a v posledním kroku, který je vidět na obrázku 16 je důležité přesunout pozici koncového bodu.



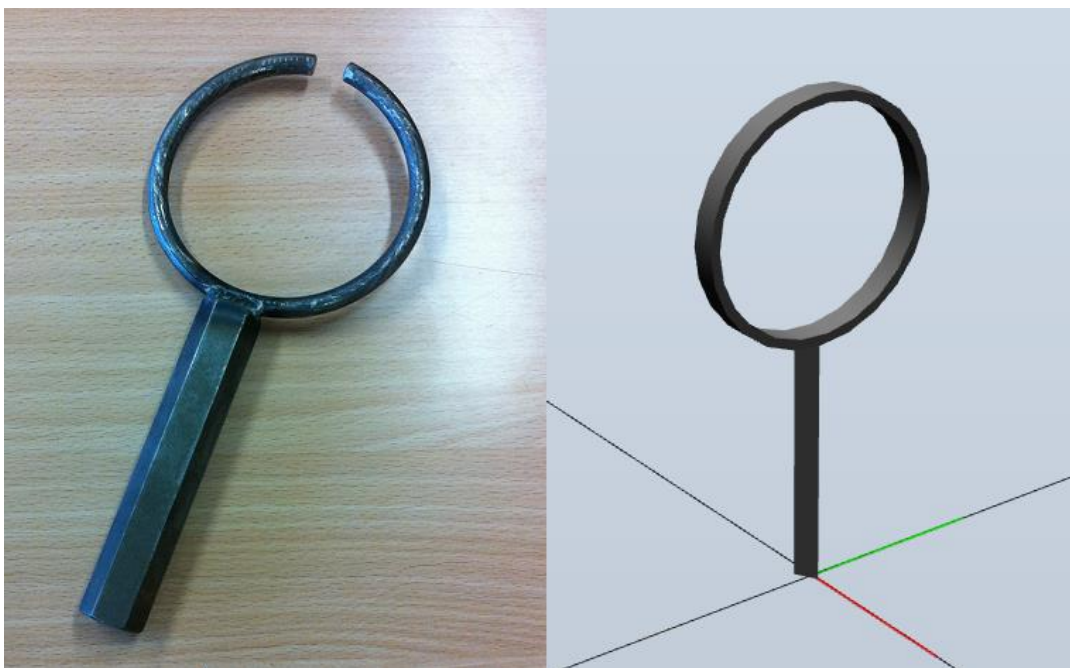
16. Poslední krok vytváření efektoru

Zbývá už jen připojit vyhotovený efektor k robotu – tím se zároveň změní pozice koncového bodu robotu na koncový bod efektoru.

#### 4.1.2 Nástroj

Jako další bylo třeba namodelovat samotný nástroj – elektrodu. Ta se skládá z několika částí, které se za pomoci použití funkcí *subtract* a *union* spojí v jeden celek. Tento model by bylo možné vytvořit i v programu AutoCAD, ale v rámci vyzkoušení a seznámení se s programem v co největším rozsahu, jsem zvolil tuto variantu. Zpětně mohu říci, že je modelování v programu RobotStudio pro toto zadání dostačující, avšak při složitějších konstrukcích je rozhodně lepší volbou AutoCAD.

Mezi namodelovanou podobou nástroje a skutečnou podobou nástroje nalezneme několik rozdílů. Jelikož, jak již bylo dříve zmíněno, pro simulaci není potřeba naprosto totožného vzhledu, ale stačí, když souhlasí rozměry. Rozdíly, které níže popíši, můžete vidět na obrázku 17.



17. Rozdíl mezi modelem a skutečnou podobou nástroje

- Model postrádá výřez, který je ve výrobku naprostou nutností, aby nemusela s nástrojem pro uvedení na trasu manipulovat obsluha
- Oko výrobku je vytvořeno z kulatiny, kdežto u modelu je vytvořeno pomocí dvou válců různých průměrů o malé výšce
- V poslední řadě je rukojeť tvořena pravidelný šestibokým hranolem, ale rukojeť modelu je vytvořena pomocí trojbokého hranolu. Tento rozdíl je dán faktem, že je trojboký hranol složitě k sehnání. Zároveň šestiboký hranol neubírá na schopnosti nástroje uchovat stálou pozici v chapadle.

#### 4.1.3 Tvorba workobjectů

V případě, že bychom potřebovali z různých důvodů měnit pozici nebo orientaci dráhy „horkého drátu“ nebo pozici elektrody, byly vytvořeny dva základní workobjecty – konkrétně: *Workobject\_Elekroda* a *Workobject\_Deska*. V případě řídicího systému IRC5 tyto dlouhé názvy neznamenaají absolutně žádný problém a můžeme je tedy použít. Pro přehlednost a programovou orientaci jsou vhodnější.

- *Workobject\_Elekroda* je Workobject, na kterém jsou umístěny targety, které souvisí především s výchozí pozicí robotu, jeho uchopením a puštění nástroje. V případě, že by odkládací plocha musela být např. z nedostatku místa posunuta,

je možné změnit pouze polohu nebo orientaci celého Workobjectu a nemusíme měnit všechny targety, jež jsou na tomto Workobjectu zahrnuty. Výchozí poloha tohoto Workobjectu je taková, aby pokud možno co nejméně překážela zbytku celé trasy. Nakonec byl celý tento Workobject přesunut tak, aby mohl být odkládací podstavec pro nástroj umístěn na podkladové desce.

- *Workobject\_Deska* je Workobject, na němž jsou umístěny targety související s celou kovovou trasou této úlohy. Tedy od najetí na začátek trasy až po vyjetí z posledního bodu trasy. Tento Workobject má stejná rozměry plochy jako skutečná velikost podkladové desky, tedy 120x50mm.

#### 4.1.4 Tvorba trasy

Nejprve byla tvořena trasa, jejíž targety obsahuje *Workobject\_Deska*. Tedy předpokládá se, že chapadlo již má ve svých čelistech uchycen nástroj – elektrodu. První a poslední target, kterých robot na tomto Workobjectu dosáhne, se vyskytují mimo celou trasu a to především proto, aby mohlo dojít k přesnému najetí na trasu. Rychlosti nájezdu a výjezdu z trasy jsou pak mnohonásobně menší než většina rychlostí v ostatním průběhu trasy.

První targety byly zvoleny tak, aby byly co nejvíce na kraji rozsahu robotu a tedy aby trasa mohla poukázat na rozsahové možnosti robotu. Stejně tak se blíží okraji rozsahu i targety poslední.

Zbytek trasy jsem se snažil tvořit členitě, nejen proto, aby byl vidět rozsah pohyblivosti os robotu, ale i proto, aby trasa vypadala efektněji pro oko diváka. Jelikož má úloha mimo jiné i zaujmout, jeví se to jako důležité. Na trase lze nalézt několik půlkruhových průběhů, ale i ostré, či tupé úhly.

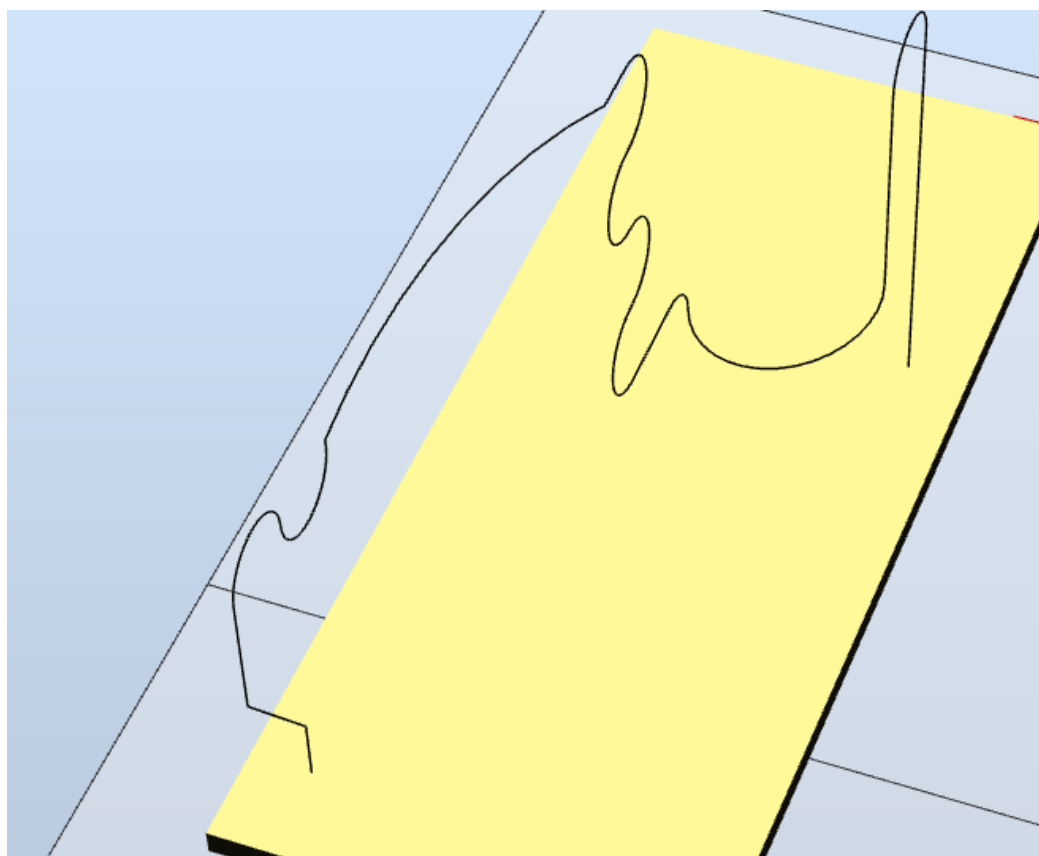
Trasa byla tedy tvořena dle vlastní fantazie, ovšem částečně byla omezena možnostmi robotu. Jednotlivé targety bylo vždy potřeba pečlivě odzkoušet i s konkrétní polohou os v daném targetu a případně trasu měnit tak, aby byl robot schopen dosáhnout požadovaných bodů nebo odpovídající orientace os.

Tvoření trasy probíhalo postupně, vždy se jednalo o navržení několika targetů, vytvoření cesty a následné odzkoušení případně změna, dokud vše neproběhlo v pořádku. Nezáleželo tedy jen na poloze targetů a orientaci os, ale i na cestě, která byla



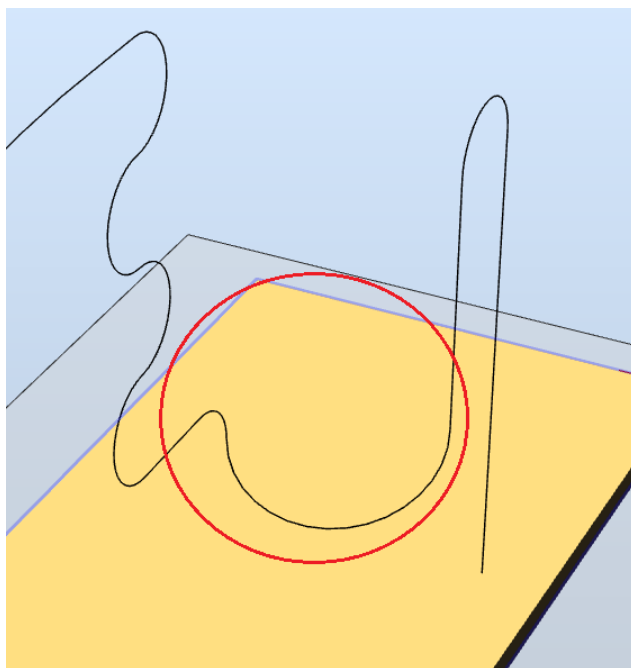
mezi targety zvolena. Občas robot dosáhne jednotlivých targetů i správné orientace, ale ne vždy je pro něj proveditelná cesta, která je mezi targety zvolena, což může způsobit velký problém a celá část trasy se musí přehodnotit a vymyslet jinak.

Po úspěšném odzkoušení celé trasy byla vytvořena její podoba pomocí modelové funkce: *Curve*  $\rightarrow$  *Line/Circle/Arc* (dle potřeby) a posléze spojena v jeden celek pomocí již dříve zmíněné funkce *Union*. Výslednou podobu namodelované trasy lze vidět na obrázku 18.



18. Namodelovaná trasa

Za nejzajímavější část trasy by se dala považovat část označená na následujícím obrázku 19, jelikož se při jejím vykonávání poměrně viditelně hýbou všechny osy robotu zároveň, čehož si pozorný divák po několikátém průběhu zajisté všimne. Zároveň se zde mění v jednom oblouku poloha koncového bodu robotu ve všech třech osách. Podobnou část, můžeme v této úloze vidět ještě v jednom místě, ovšem tam má ohyb mnohem větší rádius a pohledově se tak tato zajímavost neprojeví tak výrazně jako v tomto případě.



19. Nejzajímavější část trasy

#### 4.1.5 Uchopení a odložení nástroje

Po vyřešení průběhu celé trasy zbývalo doladit uchopení a odložení nástroje. V tomto případě jsou targety, kterých se to týká vázány k Workobjectu nazvanému: *Workobject\_Elketroda*.

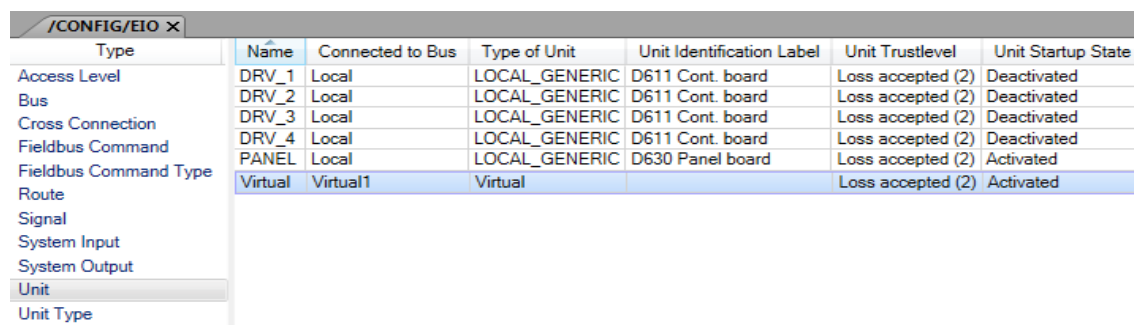
Robot se na začátku nachází ve své výchozí pozici. Posléze najede do pozice, která se na ose  $X$  a  $Z$  shoduje s polohou nástroje – do této pozice může najet v plné rychlosti, která byla pro tuto úlohu určena na  $v1000$ . Nakonec dojde k posunu na ose  $Y$  tak, aby chapadlo mohlo přesně uchopit nástroj – uchopení je popsáno v kapitole 4.1.6 – a najetí do startovní pozice. Odložení nástroje je pak obdobné, akorát v jiném pořadí, navíc při odkládání je pevně stanovena pozice na ose  $X$  a  $Y$ , při přesném dojetí k odkládacímu podstavci se mění pozice a ose  $Z$  a je vykonáváno rychlostí  $v50$ .

Pro odložení nástroje muselo být vytvořeno několik dalších targetů, aby byla přesněji vymezena cesta od koncové pozice trasy k pozici pro odložení. Pouhým příkazem *MoveJ*, bez využití dodatečných targetů, by došlo ke kolizi nástroje s dráhou.



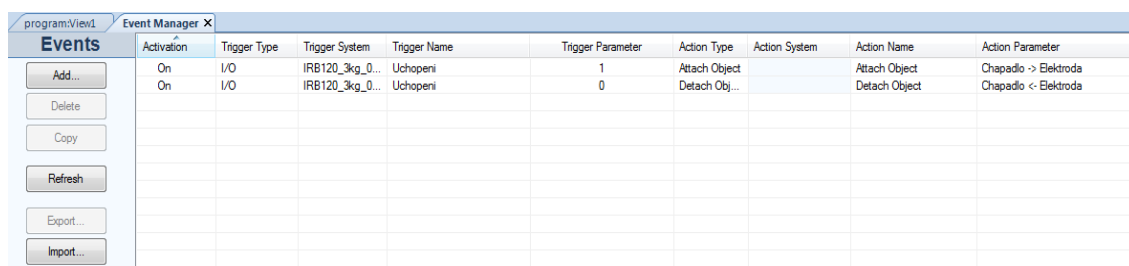
## 4.1.6 Události

Na rozdíl od reálného robotu, kde má případný použitý efektor jasně danou funkci, je potřeba ve virtuálním systému tuto funkci vytvořit a určit její přesné parametry. Tím dáme virtuálnímu kontroleru jasnou představu o tom, jaké jsou schopnosti a funkce vytvořeného efektoru a lze je tak v simulaci plně využít (předání informace na vstupy/výstupy kontroleru, uchopení předmětu chapadlem/přísavkou...) V úloze musela být nejprve vytvořena virtuální jednotka kontroleru (viz obrázek 20) dále bylo potřeba vytvořit digitální signál pro digitální simulaci reálného výstupu, kterou lze v obrázku 21 vidět jako *Uchopeni* v kolonce *Trigger Name*. Poté již mohla být vytvořena událost, která reaguje na jedničkový impulz v případě, že chceme, aby chapadlo uchopilo elektrodu. A naopak událost reagující na nulu, kdy požadujeme upuštění elektrody chapadlem – viz kolonka *Trigger Parameter*. V kolonce *Action Name* je pak uvedeno, co se má vykonat. V programu Rapid lze posléze jednoduše přidat vytvořený příkaz *Uchopeni* do místa, ve kterém je třeba nástroj uchopit/odložit. Jelikož má reálná jednotka výstup pro spuštění stlačeného vzduchu definován jako *DO00*, byl později změněn název příkazu *Uchopeni* na *DO00*, abychom při úpravě programu a opětovném nahrání do řídicího systému robotu nemusely tento název neustále měnit.



Type	Name	Connected to Bus	Type of Unit	Unit Identification Label	Unit Trustlevel	Unit Startup State
Access Level	DRV_1	Local	LOCAL_GENERIC	D611 Cont. board	Loss accepted (2)	Deactivated
Bus	DRV_2	Local	LOCAL_GENERIC	D611 Cont. board	Loss accepted (2)	Deactivated
Cross Connection	DRV_3	Local	LOCAL_GENERIC	D611 Cont. board	Loss accepted (2)	Deactivated
Fieldbus Command	DRV_4	Local	LOCAL_GENERIC	D611 Cont. board	Loss accepted (2)	Deactivated
Fieldbus Command Type	PANEL	Local	LOCAL_GENERIC	D630 Panel board	Loss accepted (2)	Activated
Route	Virtual	Virtual1	Virtual		Loss accepted (2)	Activated
Signal						
System Input						
System Output						
Unit						
Unit Type						

20. Vytvořená virtuální jednotka



Activation	Trigger Type	Trigger System	Trigger Name	Trigger Parameter	Action Type	Action System	Action Name	Action Parameter
On	I/O	IRB120_3kg_0...	Uchopeni	1	Attach Object		Attach Object	Chapadlo -> Elektroda
On	I/O	IRB120_3kg_0...	Uchopeni	0	Detach Obj...		Detach Object	Chapadlo <- Elektroda

21. Event Manager k úloze 1

#### 4.1.7 Dokončení programu

V částech trasy, kde se nevyskytuje pohyb po přímce či kružnici, je využito příkazů *MoveJ*. Tam kde se jedná o pohyb po přímce příkazů *MoveL*. Tyto dva příkazy byly používány pro tvorbu cesty pomocí funkce *Path*. V případě pohybu kruhového a tedy příkazu *MoveC* byl tento pohyb deklarován v programu Rapid přímo. Vzájemné propojení těchto částí je v programu RobotStudio řešena jednoduchou synchronizací virtuálního kontroleru a programovacím prostředím RobotStudio. Před samotným uchopením a odložením nástroje bylo přidáno čekání po dobu dvou sekund, díky čemuž je nástroj před uchopením/po odložení v klidovém stavu.

#### 4.1.8 Zkouška simulace

Po odzkoušení jednotlivých kroků byla simulace to poslední, co zbývalo u této úlohy otestovat v programu RobotStudio. Celý program byl tedy synchronizován s virtuálním kontrolérem a odzkoušen v záložce simulace. Tato konečná simulace již proběhla bez problémů. Ty však nastaly při zkoušce na reálném robotu, které způsobovaly příkazy *MoveC* a musely tedy být nahrazeny za pomoci interpolační metody – více v kapitole 5.2).

### 4.2 Úloha 2: Předání míčku

Po zralé úvaze jsem usoudil, že by bylo dobré realizovat v prostředí RobotStudio i tuto úlohu. Nespornou výhodou tohoto způsobu programování je především skutečnost, že není potřeba být fyzicky tam, kde je robot. Nahrání do robotů ABB IRB 140 a ABB IRB 1400 využívají řídicí systém S4C plus, je sice náročnější než v případě robotu ABB IRB 120, který má řídicí systém IRC5, ale i přesto má tento způsob programování nesporné výhody. Robot ABB IRB 1400 s řídicím systémem S4C nebyl použit s ohledem na komunikaci (viz výše v kapitole 2.2) a celkové náročnosti na úpravu programu z RobotStudia pro jeho použití v robotu.

#### 4.2.1 Efektory

Tvorba efektorů pro tuto úlohu probíhá totožně, jako při úloze první. Jedná se tedy jen o model, který má stejné rozměrové parametry, nikoli o naprosto přesný model

jejich podoby. Rozdílná je samozřejmě velikost efektoru a především pozice koncového bodu, jelikož přísavky, které jsou v laboratoři dostupné, mají svůj koncový bod mimo osu souměrnosti. Programová změna této osy se provádí při vytváření nástroje změnou pozice v posledním kroku. Ostatní postup při tvorbě efektorů a jejich připojení je stejný – z modelů jsou tedy funkcí *Create tool* vytvořeny efekторы, které se připojí ke koncovým členům robotů.

#### 4.2.2 Workobjecty

V této úloze jsou Workobjecty hned čtyři – a to dva, pro každého z užitých robotů. S ohledem na řídicí systém S4C plus jsou jejich názvy o dost kratší, než v případě předchozí úlohy, u které je použit robot ABB IRB 120 a tedy řídicí systém IRC5. Jedná se o následující Workobjecty: *Wo140*, *Wo1400*, *WoP140* a *WoP1400*, z nichž *Wo140* a *Wo1400* se týkají targetů určených pro cestu k místu předání pro oba roboty a zpět, kdežto *WoP140* a *WoP1400* jsou vyhrazeny pro počáteční a koncovou fázi použití obou přísavek, tedy vzetí resp. odložení míčku a to v případě každého z robotů samostatně. Použití čtyř Workobjectů může mít nesporný vliv při samotném doladění úlohy. V případě nepřesně změřené pozice robotů nebo mírného přesunu jejich reálné pozice v laboratoři robotiky pouze upravíme pozici jednotlivých Workobjectů a úloha může být bez velkých příprav vykonána.

#### 4.2.3 Tvorba cesty

Jelikož nebylo vytvoření samotné cesty tak náročné, jako v případě první úlohy, byla cesta vytvářena postupně, tak jak je vykonávána. Robot ABB IRB 140 nejprve najede do pozice vhodné pro uchopení míčku a posléze jej uchopí. Následně oba roboty dojedou do pozice pro výměnu – nejprve IRB 140 a posléze IRB 1400 tak, aby nedošlo ke kolizi. Dojde k pomalému přiblížení přísavky, která nedrží míček a jeho výměně během současné rotace obou robotů. Robot ABB IRB 140 najede do výchozí pozice přímo, robot ABB IRB 1400 dojedou do pozice pro odložení míčku. Po odložení dojedou též do výchozí pozice. Výměna se nakonec zopakuje v opačném gardu, tentokrát najede do pozice nejprve IRB 1400 a poté IRB 1400. Kromě přesné výměny a přesného najetí pro chycení a odložení míčku není potřeba robota nijak rychlostně omezovat – je dostačující, aby zvolil nejrychlejší cestu k následujícímu targetu.

Nejnáročnější částí této cesty je vzájemná výměna míčku, jelikož jsou oba roboty nejvíce přiblíženy a je tedy třeba dávat pozor, aby nedošlo k nechtěné deformaci míčku, který je v tu chvíli mezi oběma roboty. Zároveň se jedna přísavka vypíná a druhá spíná, aby byl míček vždy držen alespoň jednou přísavkou. V této úloze není potřeba používat příkazů pro kruhový pohyb – *MoveC*. Vystačíme si s příkazy *MoveJ* a *MoveL*.

Méně náročné pak bylo chycení míčku, kde je důležité najíždět vždy přesně z vrchu, stejně tak jako v případě jeho odložení. Pro toto najetí a odjetí je rychlost robotu značně snížena a je použit lineární pohyb.

#### **4.2.4 Uchopení a odložení míčku**

Targety, určené pro uchopení a odložení míčku jsou v případě robotu IRB 140 vázány na Workobject *WoP140* a v případě robotu IRB 1400 pak k Workobjectu *WoP1400*.

V případě této úlohy najíždí přísavka pro míček vždy shora, stejně tak, jako při jeho odložení. Najede tedy ve standardní rychlosti použité ve zbytku úlohy do pozice nacházející se nad tou odkládací a pomalejší rychlostí dojede přesně na určenou pozici pro vzetí/odejmutí. Souřadnice na osách X a Y zůstávají při dojezdu neměnné, mění se vždy pouze pozice v ose Z.

#### **4.2.5 Předání míčku**

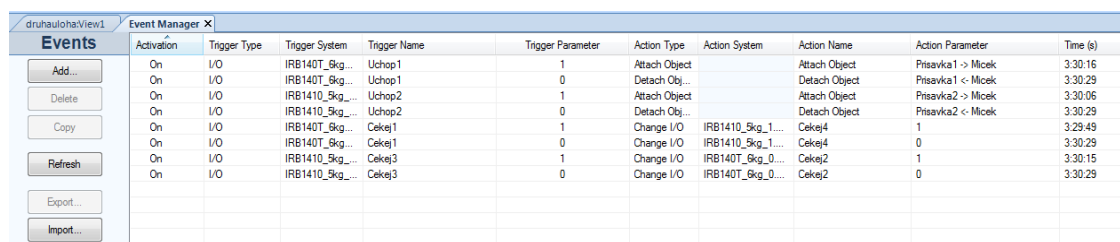
Targety související se vzájemným předáním míčku jsou vázány na Workobject *WoI40* v případě robotu IRB 140 a *WoI400* v případě robotu IRB 1400.

Přísavka jednoho z robotů v tomto případě docílí určeného targetu lehce posunutého od targetu pro předání, kam posléze dojede pomalým lineárním pohybem – mění se pouze pozice na ose Y. Přísavka druhého z robotů najíždí do pozice stejně, pouze z opačné strany a tedy i pozice na ose Y se mění opačným směrem.

## 4.2.6 Události

Stejně tak, jako v první úloze byla nejprve vytvořena digitální simulace reálného výstupu, *Uchop1* a *Uchop2* a také události pro předání informací mezi roboty *Cekej1*, *Cekej2*, *Cekej3* a *Cekej4* – ty mají za úkol informovat o tom, zda je robot připraven v pozici pro předání míčku či nikoli a na jejich základě pokračuje vykonávání programu. Tyto události jsou vidět na obrázku 22.

Poté již byly vytvořeny události, které reagují na jedničkový impulz v případě, že chceme, aby *prisavka1* resp. *prisavka2* uchopila míček. A naopak událost reagující na nulu, kdy požadujeme upuštění míčku přísavkou. V programu Rapid posléze jednoduše přiřadíme příkazy pro uchopení a čekání do míst, na které je třeba.



Activation	Trigger Type	Trigger System	Trigger Name	Trigger Parameter	Action Type	Action System	Action Name	Action Parameter	Time (s)
On	I/O	IRB140T_9kg...	Uchop1	1	Attach Object		Attach Object	Prisavka1 -> Micek	3:30:16
On	I/O	IRB140T_9kg...	Uchop1	0	Detach Obj...		Detach Object	Prisavka1 < Micek	3:30:29
On	I/O	IRB1410_5kg...	Uchop2	1	Attach Object		Attach Object	Prisavka2 -> Micek	3:30:06
On	I/O	IRB1410_5kg...	Uchop2	0	Detach Obj...		Detach Object	Prisavka2 < Micek	3:30:29
On	I/O	IRB140T_9kg...	Cekej1	1	Change I/O	IRB1410_5kg_1...	Cekej4	1	3:29:49
On	I/O	IRB140T_9kg...	Cekej1	0	Change I/O	IRB1410_5kg_1...	Cekej4	0	3:30:29
On	I/O	IRB1410_5kg...	Cekej3	1	Change I/O	IRB140T_9kg_0...	Cekej2	1	3:30:15
On	I/O	IRB1410_5kg...	Cekej3	0	Change I/O	IRB140T_9kg_0...	Cekej2	0	3:30:29

22. Event Manager k úloze 2

## 4.2.7 Dokončení programu

Zbytek, je opět řešen obdobně jako v předchozí úloze. Při dojezdech k převzetí míčku jsou používány oproti zbytku programu nízké rychlosti, aby vše proběhlo v pořádku. Zároveň je používána funkce *MoveL* a po přesném najetí do pozice pro převzetí probíhá čekání. Čekání je též zařazeno do části programu před výměnou míčku a po ní.

## 4.2.8 Zkouška simulace

V programu RobotStudios už scházela jen samotná simulace. Po odzkoušení jednotlivých kroků v tom nebyl problém. Celý program byl tedy synchronizován a odzkoušen. Konečná simulace proběhla bez jakýchkoli komplikací. Její průběh lze opět shlédnout na přiloženém CD.

## 5 Konečné doladění

### 5.1 Aretační systém a ukotvení

Aretační systém je v tomto případě řešen jednoduše – tedy mechanickým ukotvením v podobě dvou velkých šroubů se zápusnou hlavou a matkou. Jelikož není trasa ani podstava, na které drží symetrická, není potřeba více, než dvou šroubů.

Po přesném umístění trasy byl provrtán její podstavec i stůl, na kterém je umístěn robot zároveň, jelikož bylo potřeba přesně zachovat vyhrazenou pozici.

Pokud bychom chtěli, aby celý tento podstavec byl odpružen, výpočet koeficientu pružnosti pružiny počítán naprosto stejně jako při výpočtu pro podstavec určený k odkládání nástrojů – viz vzorec 11.

Aretace nástroje je vyřešena jeho vytvarováním, je též lehce řešena u odkládacího podstavce pro nástroj, v němž je vyryta drážka, do které po odložení nástroj přesně zapadne.

V našem případě není třeba robota přesně seřizovat či nastavovat. V případě, že potřebujeme změnit některou z poloh, stačí v programu změnit polohu jednotlivých Workobjectů.

### 5.2 Problémy

Prvním z problémů, které se vyskytli, se týkal doladění úlohy „Horký drát“. Vzhledem ke skutečnosti, že byla fyzická podoba úlohy vytvářena tzv. na koleni, došlo na trase k drobným odchylkám v podobě rozměrů. Tento problém byl vyřešen změnou polohy workobjectu trasy a pár konkrétních targetů tak, aby namodelovaná trasa více odpovídala trase skutečné.

Dalším, poměrně vážnějším problémem bylo zasekávání se robotu v určitých částech trasy. FlexPendant oznámil upozornění na chybnou konfiguraci, avšak po potvrzení byl robot schopen cestu vykonat i s touto „chybnou“ konfigurací. Tato chyba pro mě byla celkem zarážející, jelikož program RobotStudio vždy při chybné

konfiguraci chybu oznámil a program se nevykonával. V tomto případě však program proběhl v RobotStudios i během simulace korektně. Chyby se vyskytovaly vždy v případech, kdy byla použita funkce pro kruhový pohyb – *MoveC*. Jedinou schůdnou metodou, jak tento problém odstranit bylo přepracování celé trasy a využít funkci *AutoPath*, která využívá v našem případě lineární aproximaci. Vzhledem k tomu, že již byla celá trasa namodelovaná, bylo možné tuto funkci bez větších problémů využít. Poté zbývalo pouze změnit orientaci veškerých targetů, odzkoušet konfiguraci robotu, simulaci a v poslední řadě vyzkoušet celý upravený program v reálném robotu. Došlo též k úpravě zón v průběhu celé trasy, aby byl celý tento průběh plynulejší. Zóny byly změněny na z5. Tento program proběhl již absolutně bez komplikací. Simulaci můžete shlédnout na přiloženém CD.

Poslední chybou, na kterou jsem narazil, bylo natočení efektoru reálného robotu o  $10^\circ$  v ose Z. Tato chyba byla odstraněna změnou orientace efektoru v programu RobotStudio totožným způsobem.

## Závěr

Cílem mé bakalářské práce bylo vytvořit demoúlohy pro potřeby prezentace průmyslových robotů ABB, přičemž jedna z úloh měla být vytvořena v prostředí RobotStudio. Důraz byl kladen též na možnost rychlé přípravy úloh na prezentaci robotů, např. na dny otevřených dveří na Fakultě mechatroniky, informatiky a mezioborových studií.

Vzhledem k tomu, že jsem chtěl více poukázat též na vzájemnou spolupráci dvou robotů, rozšířil jsem práci o průmyslového robota ABB IRB 1400, který je rovněž dostupný v laboratoři robotiky na Technické Univerzitě v Liberci. K tomuto kroku jsem musel přistoupit z důvodu, že roboty ABB IRB 120 a ABB IRB 140 nejsou ve vzájemném dosahu a bez nutných zásahů týkajících se rozmístění zmíněných robotů v laboratoři by nebylo možné úlohu realizovat. Je tedy mnohem efektivnější použít k této spolupráci robot ABB IRB 1400 jak zmiňuji v textu této práce.

Při tvorbě úlohy „Horký drát“, kterou jsem vytvářel v programu RobotStudio mě jeho nesporné výhody zaujali natolik, že jsem se rozhodl pro jeho plné využití i v programu „výměna míčku“. V případě, že by v pozdější době došlo k přestavbě laboratoře a změnila by se pozice robotů, lze u obou úloh využít softwareové korekce právě v programu RobotStudio – změnou pozice workobjectů – a úlohy opět bez větších zásahů používat. Úloha „Horký drát“ je kompletně odzkoušena a provozuschopná. Úlohu „předání míčku“ se z časových důvodů nepodařilo odzkoušet na reálných robotech.

Jelikož je laboratoř během roku využívána pro výuku předmětů týkajících se robotiky, bylo nezbytně nutné zajistit snadnou demontáž a montáž úloh. V případě trvalé montáže úlohy „Horký drát“ by došlo ke značnému znemožnění práce s robotem ABB IRB 120, což je z pochopitelných důvodů nepřijatelné. Úloha je tedy koncipovaná tak, že ji zvládne obsluha (jedna osoba) bez větších problémů sama nainstalovat během velice krátké doby. Aretační systém, který je v této úloze vytvořen, zajišťuje dostatečně přesnou pozici a tak není třeba softwareové korekce – či jakéhokoli jiného zásahu obsluhy do programu – aby mohla být úloha spuštěna a vykonána (vyjma případu zmíněného v předchozím odstavci). Deska se pouze usadí do připravené pozice a zapojí se obvod pro detekci dotyku. Zajímavým rozšíření této práce by bylo využití konektoru



s digitálními vstupy a implementované kabeláže robotu IRB 120 namísto volně vedeného drátku použitého k propojení obvodu pro detekci s elektrodou. Případně obvod upravit tak, aby kontrolu dotyku zajišťoval řídicí systém robotu a v případě dotyku např. přerušit program.

Nejsem si jistý, zda dokáží studenti ocenit přínos této práce při své výuce, ale pevně věřím, že přispěje Technické Univerzitě v Liberci k ještě většímu zájmu o studium na Fakultě mechatroniky, informatiky a mezioborových studií a nechá také nahlédnout do zajímavé a perspektivní oblasti průmyslu. Zároveň doufám, že bude motivující pro ostatní alespoň tak, jako byla pro mě.

## Literatura

- [1] Technická dokumentace ABB RapidLanguage, ABB RobotStudio
- [2] Skařupa J. *Průmyslové roboty a manipulátory*, VŠB-TUO Ostrava 2007. ISBN 978-80-248-1522-0
- [3] GREPL, Robert. *Kinematika a dynamika mechatronických systémů*. v Brně: CERM, 2007. ISBN 978-80-214-3530-8
- [4] KLÍMA, Jan. ÚŘADU PRO TECHNICKOU NORMALIZACI, metrologii a státní zkušebnictví. *Roboty a robotická zařízení – Slovník*. 2. vyd. Praha: Svaz strojírenské technologie, 2013. ČSN ISO 8373 18 6501
- [5] KARGER, Adolf a Marie KARGEROVÁ. *Základy robotiky a prostorové kinematiky*. Praha: ČVUT, 2008. ISBN 80-01-02183-1
- [6] ZÁDA, Václav. TECHNICKÁ UNIVERZITA V LIBERCI. *Roboty na Fakultě mechatroniky, informatiky a mezioborových studií* [online]. v Liberci, 15.02.2011 [cit. 2014-05-16]. Dostupné z: [http://www.fm.tul.cz/~vaclav.zada/Laborator\\_Robotiky/Roboty%20na%20MTI-Zada.pdf](http://www.fm.tul.cz/~vaclav.zada/Laborator_Robotiky/Roboty%20na%20MTI-Zada.pdf)
- [7] ABB. *IRB 120 data sheet* [online]. 2012 [cit. 16.05.2014]. Dostupné z: [http://www05.abb.com/global/scot/scot241.nsf/veritydisplay/3bd625bab3c7cae1c1257a0800495fac/\\$file/ROB0149EN\\_D\\_LR.pdf](http://www05.abb.com/global/scot/scot241.nsf/veritydisplay/3bd625bab3c7cae1c1257a0800495fac/$file/ROB0149EN_D_LR.pdf)
- [8] ABB. *IRB 140 data sheet* [online]. 2003 [cit. 16.05.2014]. Dostupné z: [http://www05.abb.com/global/scot/scot241.nsf/veritydisplay/98ba43a906331fec48257c6f00374818/\\$file/PR10031EN%20R15\\_En.pdf](http://www05.abb.com/global/scot/scot241.nsf/veritydisplay/98ba43a906331fec48257c6f00374818/$file/PR10031EN%20R15_En.pdf)
- [9] ABB. *IRB 1400 data sheet* [online]. 2005 [cit. 16.05.2014]. Dostupné z: [http://www05.abb.com/global/scot/scot230.nsf/veritydisplay/99bb3fb8ff6495cfc1257b130056d120/\\$file/IRB1400\\_R3-US%2002\\_05.pdf](http://www05.abb.com/global/scot/scot230.nsf/veritydisplay/99bb3fb8ff6495cfc1257b130056d120/$file/IRB1400_R3-US%2002_05.pdf)
- [10] ABB S4C [online]. [cit. 16.05.2014]. Dostupné z: <http://www.used-robots.com/abb/controller/s4c>

- [11] S4Cplus *data sheet* [online]. 2000 [cit. 16.05.2014]. Dostupné z:  
[http://www02.abb.com/global/nlabb/nlabb034.nsf/bf177942f19f4a98c1257148003b7a0a/068b10a5e5aa6ab1c1256f5b0047e00c/\\$FILE/Datasheet%20S4Cplus.pdf](http://www02.abb.com/global/nlabb/nlabb034.nsf/bf177942f19f4a98c1257148003b7a0a/068b10a5e5aa6ab1c1256f5b0047e00c/$FILE/Datasheet%20S4Cplus.pdf)
- [12] IRC5 *data sheet* [online]. 2011 [cit. 16.05.2014]. Dostupné z:  
[http://www05.abb.com/global/scot/scot241.nsf/veritydisplay/c13e1c5490c61230c125796000515137/\\$file/IRC5%20datasheet%20PR10258%20EN\\_R13.pdf](http://www05.abb.com/global/scot/scot241.nsf/veritydisplay/c13e1c5490c61230c125796000515137/$file/IRC5%20datasheet%20PR10258%20EN_R13.pdf)
- [13] Obrázek,  
[http://www.festo.com/PDF\\_Flip/trends\\_in\\_automation/1\\_2012/CZ/files/assets/seo/page16.html](http://www.festo.com/PDF_Flip/trends_in_automation/1_2012/CZ/files/assets/seo/page16.html)
- [14] Obrázek, <http://www.conrad.at/ce/de/product/502802/Zubehoer-fuer-Serie-40-41-44-Serie-95-Finder-9505-Schraubfassung-mit-Schnappbefestigung-fuer-DIN-Schiene-Sichere-Tren>
- [15] Obrázek,  
<http://www.soselectronic.cz/?str=371&artnum=6673&name=finder-40-52-8-230-0000>
- [16] Obrázek, [http://www.festo.com/cms/cs\\_cz/15573.htm](http://www.festo.com/cms/cs_cz/15573.htm)

## **A Obsah CD**

- /Text Bakalářské Práce – obsahuje textovou formu bakalářské práce ve formátech .docx a .pdf
- /Video – obsahuje záznamy simulací obou úloh a video reálného provedení úlohy Horký drát
- /RobotStudio – obsahuje veškeré potřebné a použité soubory a knihovny potřebné pro spuštění či úpravu úloh